ELSEVIER

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 16 (2013) 89 – 97

Procedia
Computer Science

Conference on Systems Engineering Research (CSER'13)
Eds.: C.J.J. Paredis, C. Bishop, D. Bodner, Georgia Institute of Technology, Atlanta, GA, March 19-22, 2013.

Network Science Enabled Cost Estimation in Support of MBSE

Matthew Dabkowski[a*], Jose Estrada[a], Ben Reidy[a], Ricardo Valerdi[a]

[a] *University of Arizona, 1127 E. James E. Rogers Way, Tucson, AZ 85721-0012, USA*

Abstract

As Model-Based Systems Engineering (MBSE) continues to mature, additional analyses can be performed to understand the relationships between a system's cost, schedule, and performance. The DoD Architecture Framework (DoDAF) is one of many approaches for representing the relationship between system components. If DoDAF diagrams - such as those provided in the Systems View - are treated as networks, they can be mathematically analyzed, providing additional summary metrics that might yield valuable insight into the degree of effort (i.e., cost) required to bring a system to fruition. For instance, if we view the addition of a new subsystem to the system's architecture abstractly as the addition of a vertex to a network, we can apply contemporary methods from network science to grow the network and estimate its cost. Unlike rules of thumb, this approach provides us with an objective way to quantify and assess change. The objective of this work is to turn MBSE knowledge into computational knowledge in order to support tradeoffs, change impact analysis, and related approaches early in the life cycle.

Keywords: Model Based Systems Engineering; DoD Architecture Framework; COSYSMO; network science

1. Introduction

Traditional approaches to systems engineering utilize documents as the authoritative source of information concerning the system's requirements, constraints, architecture, design, trade-offs, decisions, and other pertinent information. While this approach has served the profession well, an unfortunate characteristic of document-based systems engineering is the apparent disconnect between the elements within a given document and other documents. In particular, if a change in one document necessitates a change(s) in others, these adjustments must be manually made. In this environment, coordination within and between engineering teams is paramount, and breaks in communication inevitably introduce inconsistencies and errors.

In order to make the system development process more dynamic, this paper presents a methodology that utilizes the information in DoDAF diagrams to perform more sophisticated analyses in less time. Specifically, we focus on integrating a cost model with DoDAF diagrams to explore the cost impact of architectural changes.

* Corresponding author. Tel.: 520-248-6130; fax: 520-621-6555.
E-mail address: mfd1@email.arizona.edu.

doi:10.1016/j.procs.2013.01.010

2. Model-Based Systems Engineering

The systems engineering profession is currently transitioning from a document-based to a model-based approach. Model-Based Systems Engineering (MBSE) is defined by the International Council on Systems Engineering (INCOSE) as the "formalized application of modeling to support system requirements, design, analysis, verification, and validation, beginning in the conceptual design phase and continuing throughout development and later life cycle phases" [1]. With the introduction of MBSE, project information can be readily shared within large, complex projects; changes can be easily accommodated; and comprehensive traceability can be automated. In essence, MBSE provides the engineering team with an effective, efficient, and consistent framework for the entire project, while maintaining the necessary flexibility to tailor individual projects or address special circumstances. The net effects of MBSE's benefits are improved quality and productivity with lower risk.

2.1. State of the art

Today MBSE is commonly practiced across industry due to the availability of standards-based MBSE tools, mature modeling methods, and enhanced integration between architecture models, simulation, and behavior analysis. These state of the art MBSE tools vary in price and capability; however, most can link all of the elements of a particular project into a central model which provides enhanced visibility of risks, design weaknesses, and avoidable expenses. With MBSE tools, the engineering team can identify requirement issues early in the life cycle, improve impact analysis of requirements changes, and conduct early/on-going requirements validation and verification. In fact, savings from MBSE have been demonstrated in a variety of contexts. Westinghouse saved 70% on verification using auto-generated testing for railway switching systems [2]. Raytheon found a 68% reduction in specification defects since MBSE practices were introduced [3].

2.2. Opportunities

Despite the numerous advantages of MBSE over the traditional document-based approach, there are many opportunities for improvements. For example, transferring from static to dynamic models offers the potential to facilitate a quantitative analysis of requirements, interfaces, architectures, functionality, tradeoffs, and test and evaluation. There is also an opportunity to incorporate cost modeling into MBSE, augmenting the engineering team's ability to understand the cost of architectural changes and to perform tradeoff analyses.

3. DoDAF and Parametric Cost Modeling

3.1. Filling the void

The size and complex nature of DoD procurements requires analyses to be done using frameworks such as the DoD Architecture Framework (DoDAF). Now that DoDAF is in its second generation [4], vendors like Atego, IBM, Mega, NoMagic, Sparx Systems and Vitech have created sophisticated modeling tools [5] where the various views can be implemented into an interactive design and development environment which we refer to as MBSE. DoDAF is now mandated for all programs of record [4] and will soon be adopted by all 37 departments of the federal government.

Despite the increased use of DoDAF as a foundation for MBSE, a void remains between evaluating technical trades and assessing their costs. The current method for determining cost in MBSE is bottom-up which involves adding the individual components to determine the total cost. This has two drawbacks: (1) it is difficult to perform a bottom-up estimate early in the life cycle because the maturity of the design definition limits the specificity of subsystem and component costs, and (2) this method tends to be more time consuming and infeasible when time or personnel are limited [6].

To address this gap, we propose an extension to an existing MBSE environment by incorporating a parametric cost model to facilitate "should cost/will cost" analysis. In particular, a complement to bottom-up estimation is parametric estimation, also known as the top-down approach. This method is based on the overall characteristics of the project and is more applicable to early cost estimates when only global properties are known [7]. More importantly, it is based on a cost estimation relationship (CER) using organizational and technical characteristics of a project to determine its cost. Using parametric models also helps establish cost estimates as a function of historical performance, a shortfall often cited as a common cause for poor performance of large programs [8, 9].

The first step involves the implementation of a cost model into the MBSE environment. Since the objective here is to provide a proof-of-concept, we will use the open academic cost model called COSYSMO (Constructive Systems Engineering Cost Model). This model has been used in various organizations [10, 11] and has the following cost estimating relationship:

$$PM_{NS} = A \cdot \left(\underbrace{\sum_{k=1}^{4} w_{e_k}\Phi_{e_k} + w_{n_k}\Phi_{n_k} + w_{d_k}\Phi_{d_k}}_{\text{size}} \right)^{E} \cdot \underbrace{\prod_{j=1}^{14} EM_j}_{\text{effort}} \tag{1}$$

where . . .

PM_{NS} = system engineering effort in nominal months,
A = calibration constant derived from historical project data,
w_{ik} = weight for the i^{th} complexity level of the k^{th} size driver ($i \in$ {easy (e), nominal (n), difficult (d)}),
Φ_{ik} = quantity of k^{th} size driver with complexity level i ($k \in$ {1 (requirements), 2 (interfaces), 3 (algorithms) and 4 (operational scenarios)}),
E = diseconomies of scale constant, and
EM_j = systems engineering effort multiplier for the j^{th} cost driver (with a default value of 1).

As indicated in (1) above, COSYSMO's CER incorporates both the size of the system and the effort required to bring it to fruition. For additional information regarding COSYSMO, the interested reader should consult [10].

3.2. Connecting COSYSMO and DoDAF

This paper proposes a link between the COSYSMO CER and specific DoDAF views that impact systems engineering cost. Of the 60 possible combinations of DoDAF views, there are thousands of possible interdependencies. The obvious ones are "vertical" interdependencies along each principal thread. For example, one system view that is critical is the SV2 Systems Resource Flow Description because it captures the type of information exchanged between subsystems. From a cost estimation standpoint, the SV2 diagram provides not only the number of interfaces but also their type and complexity. Using this information, the analyst can evaluate the system's interfaces and assign relative weights according to the rating scale in Table 1 below.

Table 1. COSYSMO rating scale and relative weights [10].

Characteristics	Easy	Nominal	Difficult
Message complexity	Simple	Moderate	Complex
Coupling level	Uncoupled	Loose	Tight
Stakeholder consensus	Strong	Moderate	Low
Behavior	Well behaved	Predictable	Emergent
Relative weight (w_{ik})	1.1	2.8	6.3

Moreover, changes made to the SV2 diagram will propagate to the SV3 Systems-Systems Matrix, a "useful tool for managing the evolution of solutions and infrastructures, the insertion of new technologies and functionality, and the redistribution of systems and activities in context with evolving operational requirements" [12].

3.3. A simple example

With the SV2 – SV3 interdependency in mind, suppose we are in the early stages of the systems development life cycle, and we have progressed far enough that we have a draft SV2 diagram and preliminary estimates for several of

COSYSMO's drivers. Suppose that the system specification from our principal stakeholder identifies 200 easy, 200 nominal, and 50 difficult requirements, as well as 5 difficult critical algorithms. Moreover, the SV3 depicted in Figure 1 below shows that we have 10 subsystems {A, B, . . . , J} with 14 interfaces, and our analysis of the SV2 established the complexity distribution of these interfaces as 9 easy, 3 nominal, and 2 difficult.

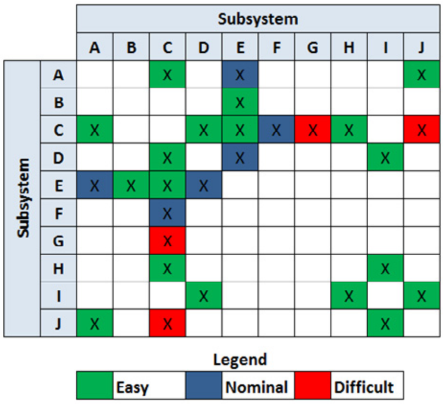


Fig. 1. SV3 for a hypothetical system.

Using A = 0.25, E = 1.06, and the product of the effort multipliers as 0.89 (based on historical data and information obtained from experts), we apply equation (1) to obtain an initial estimate of the PM_{NS} as follows (Note: Additional w_{ik} and EM_j data obtained from [10]):

$$PM_{NS} = 0.25 \cdot \left(\underbrace{(0.5 \times 200 + 1.0 \times 200 + 5.0 \times 50)}_{\text{requirements}} + \underbrace{(11.5 \times 5)}_{\text{algorithms}} + \underbrace{(1.1 \times 9 + 2.8 \times 3 + 6.3 \times 2)}_{\text{interfaces}} \right)^{1.06} \cdot 0.89 = 209.28$$

The 209.28 Person Month effort estimate can be converted into dollars by multiplying by the monthly rate in each organization. More importantly, the cost impact of the elimination or introduction of new subsystems and interfaces can be easily quantified since there is an explicit connection between the SV3 diagram and the cost model.

3.4. Research questions

Based on our previous discussion and the example above, we posit the following research questions:

- Can parametric cost estimation adequately capture the monetary impact of architectural changes early in the system lifecycle?
- Can parametric cost estimation models be embedded within MSBE software to generate credible, on-the-fly estimates?
- Does the marriage of parametric cost modeling and MBSE improve architectural decision-making, and, ultimately, reduce cost, speed-up delivery, and reduce risk?

Of course, answering these questions will require extensive research well beyond the scope of this paper. Nonetheless, the first query motivates the others, and it is amenable to exploratory investigation. As such, the remainder of this paper will focus on quantifying the complexity of architectures.

4. Modeling the system as a network

Although the formal, mathematical analysis of graphs dates to the early 18th century, the advent of the modern computer, coupled with the recent popularity and proliferation of social networks, has transformed it from an esoteric endeavor into the burgeoning, interdisciplinary field of network science [13]. At its core, network science is concerned with modeling and analyzing the connections (or edges) that exist between a network's components (or vertices), and its application to MBSE seems both obvious and natural, as demonstrated recently in a number of publications [14, 15, 16]. Consider our example system above. If we view the system's subsystems as vertices and its interfaces as edges, then (as seen in Figure 2a) the SV3 diagram shown in Figure 1 is nothing more than an adjacency matrix (***A***) for an undirected, unweighted graph (***G***) with 10 vertices (***V***) and 14 edges (***E***). Moreover, if we incorporate the interface complexity information from our earlier COSYSMO estimate, we can easily transform ***A*** into a weighted adjacency matrix (***W***) by replacing $a_{ij} = 1$ with the interface's complexity level weighting (w_{ij}) (see Figure 2b below). Represented in this way, our system can be imported into network analysis software (such as R's *igraph* package) for a detailed mathematical examination [17].

	A	B	C	D	E	F	G	H	I	J
A	0	0	1	0	1	0	0	0	0	1
B	0	0	0	0	1	0	0	0	0	0
C	1	0	0	1	1	1	1	1	0	1
D	0	0	1	0	1	0	0	0	1	0
E	1	1	1	1	0	0	0	0	0	0
F	0	0	1	0	0	0	0	0	0	0
G	0	0	1	0	0	0	0	0	0	0
H	0	0	1	0	0	0	0	0	1	0
I	0	0	0	1	0	0	0	1	0	1
J	1	0	1	0	0	0	0	0	1	0

	A	B	C	D	E	F	G	H	I	J
A	0	0	1.1	0	2.8	0	0	0	0	1.1
B	0	0	0	0	1.1	0	0	0	0	0
C	1.1	0	0	1.1	1.1	2.8	6.3	1.1	0	6.3
D	0	0	1.1	0	2.8	0	0	0	1.1	0
E	2.8	1.1	1.1	2.8	0	0	0	0	0	0
F	0	0	2.8	0	0	0	0	0	0	0
G	0	0	6.3	0	0	0	0	0	0	0
H	0	0	1.1	0	0	0	0	0	1.1	0
I	0	0	0	1.1	0	0	0	1.1	0	1.1
J	1.1	0	6.3	0	0	0	0	0	1.1	0

Fig. 2. (a) This matrix of 0's and 1's represents the unweighted adjacency matrix, ***A***, for our example system, where $a_{ij} = 1$ indicates the existence of an interface between subsystems *i* and *j*. Gray-shaded column and row labels have been added to facilitate reference. (b) The weighted adjacency matrix, ***W***, representation for the same system.

While the analysis possibilities are seemingly limitless, we will demonstrate an interesting, potential application through a simple scenario involving our example system. First, recall that although we have progressed far enough in our development process to produce a draft set of DoDAF diagrams and to construct an initial cost estimate using COSYSMO, we are still in the early stages of the product lifecycle. As such, the current architecture will undoubtedly change. Now, suppose we are interested in estimating the potential cost impact of adding another subsystem (K) to the architecture. Without information on the purpose or function of this subsystem, our ability to estimate this seems limited. However, if we view our system as a network, as represented by ***W***, network science provides an interesting approach.

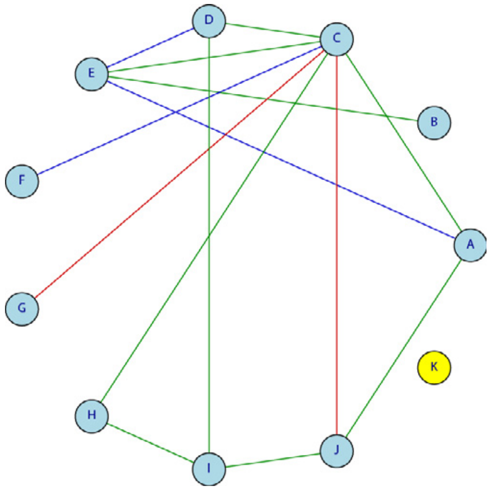
4.1. Growing the system

The problem above is represented visually in Figure 3. Specifically, if we add vertex K (v_K, the subsystem under investigation) to the current network, how will it connect to the existing structure? Equivalently, what values should we place in the k^{th} row and column of ***W***?

To answer this question, we first observe that v_C has the greatest number of edges in the network, as it is connected to 7 of the other 9 vertices. In graph theoretic terms, v_C has degree (d) 7. Intuitively, if we add v_K to the network, the likelihood of it attaching to v_C seems greater than it attaching to a lesser connected vertex, say v_G with $d = 1$. This phenomenon, where highly connected vertices are more likely to receive additional edges, appears frequently in nature, and it can be expressed using the Barabási–Albert preferential attachment (PA) model. Formally, the PA model states that the probability of a new vertex attaching to an existing vertex i (p_i) is given by $p_i = d_i / \Sigma_j d_j$ [18]. Using this relation, the preferential attachment probabilities for our current network are given in Table 2 below.

Table 2. Preferential attachment probabilities.

System (i)	A	B	C	D	E	F	G	H	I	J	Sum
d_i	3	1	7	3	4	1	1	2	3	3	28
p_i	0.107	0.036	0.250	0.107	0.143	0.036	0.036	0.071	0.107	0.107	1



	A	B	C	D	E	F	G	H	I	J	K
A	0	0	1.1	0	2.8	0	0	0	0	1.1	
B	0	0	0	0	1.1	0	0	0	0	0	
C	1.1	0	0	1.1	1.1	2.8	6.3	1.1	0	6.3	
D	0	0	1.1	0	2.8	0	0	0	1.1	0	
E	2.8	1.1	1.1	2.8	0	0	0	0	0	0	
F	0	0	2.8	0	0	0	0	0	0	0	
G	0	0	6.3	0	0	0	0	0	0	0	
H	0	0	1.1	0	0	0	0	0	1.1	0	
I	0	0	0	1.1	0	0	0	1.1	0	1.1	
J	1.1	0	6.3	0	0	0	0	0	1.1	0	
K											0

Fig. 3. (a) The current system represented as an undirected, weighted graph, where the edge weights (w_{ij}) are indicated by the color of the edge (green ⇒ $w_{ij} = 1.1$, blue ⇒ $w_{ij} = 2.8$, red ⇒ $w_{ij} = 6.3$). (b) The augmented, weighted adjacency matrix $\boldsymbol{W}$.

By analogy, the PA model captures the "rich-get-richer" effect [19], and this is clearly visible in Table 2 where v_C (with 7 edges) is 7 times more likely to receive a new edge than vertices with a single edge (v_B, v_F, v_G). However, in the PA model, the number of edges for each vertex added to the network is constant and specified in advance. Translating this to our scenario, this assumption implies that we have perfect information on how many interfaces subsystem K will generate. Of course, with no information on the purpose or function of subsystem K, this is dubious at best. With this in mind, we draw on an extension of the PA model where the degree of our incoming vertex v_K is represented as a random variable, D_{vK} [19]. This provides a "rich-by-birth" effect [20], as v_K could plausibly enter the network with a large number of edges. Accordingly, as seen in Table 3 below, we modeled the probability mass function (pmf) of D_{vK} using the observed degree distribution of the current network. Implicit in using this approach is the assumption that the structure of the current network provides a reasonable facsimile of the structure of our future network (the current network after the addition of v_K). Put another way, barring a massive redesign, the existing architecture of our system bounds and governs its future architecture.

Table 3. Probability mass function for Dv_K, where d = degree and n_d = the number of vertices in the current network with degree d.

d	1	2	3	4	7	Sum
n_d	3	1	4	1	1	10
$P(D_{vK} = d)$	0.3	0.1	0.4	0.1	0.1	1

At this point, we have established simple, intuitive methods for assigning the degree of v_K and the likelihood that one of its edges will attach to an existing vertex of the network. As such, the last task is to determine how the edge weights will be assigned. Drawing on the reasoning above, one approach is to use the overall, observed weight distribution of the current network. That said, we believe this approach is overly naïve because it ignores valuable topological information. For example, as seen in Figure 3a, if subsystem K attaches to subsystem C, then it is reasonable to expect the complexity of the interface to be nominal (blue) or difficult (red). On the other hand, if subsystem K attaches to I, where each of its 3 interfaces is rated as easy (green), then assigning a large weight to this edge seems inappropriate. Simply put, the subsystem(s) that subsystem K attaches to provides us with information on the potential complexity of the attachment(s). Accordingly, we modeled the edge weight pmf of w_{iK} using a conditional, observed distribution function (i.e., P($w_{iK} = w$ | K attaches to i) for $i \in$ {A, B, . . . , J}, which, for a given i, equates to P($w_{ij} = w$) for $i, j \in$ {A, B, . . . , J}, $i \neq j$).

4.2. Simulating growth

Using the framework described above, we implemented a Monte Carlo simulation in the statistical software R to estimate the potential cost impact of adding subsystem K to the architecture. The high-level pseudo-code for this simulation is as follows:

For a specified number of iterations . . .

(1) Initialize the system as the current system
(2) Generate a realization for D_{vK} (d); this is the number of subsystems K will attach to.
(3) Connect K to d subsystems of the current system using the PA model.
(4) For each interface established in (2), assign complexity (w_{iK}).
(5) Estimate the cost for the augmented system using COSYSMO (PM_{NS}*).
(6) Calculate the additional cost of adding subsystem K (PM_{NS}* − PM_{NS}).
(7) Store results and return to (1)

The results of 10,000 iterations are summarized graphically in Figure 4. As we would expect, the relative frequencies of subsystem K attaching to the existing subsystems reflect the ordering of their preferential attachment probabilities, providing a quick, visual verification of the simulation. Moreover, subsystem K attached to subsystem C in 54.1% of the iterations, reinforcing its position as the architecture's hub.

From a cost perspective, the 95% confidence interval for the expected cost of adding subsystem K to the architecture is (1.67, 1.71) in PM_{NS}. Additionally, as seen in Figure 4a and Table 4, there is a 50% chance that the additional cost will be less than 1.36 PM_{NS} and a 90% chance it will be less than 3.55 PM_{NS}. Translated into dollars, if we accept that our current model provides a reasonable estimate of system growth, our "best guess" for the cost of adding subsystem K is $33,811 (1.69 PM_{NS} (the mean of our iterations) × $20,000 / PM_{NS}), and it should not exceed $137,000 (6.85 PM_{NS} × $20,000 / PM_{NS}).

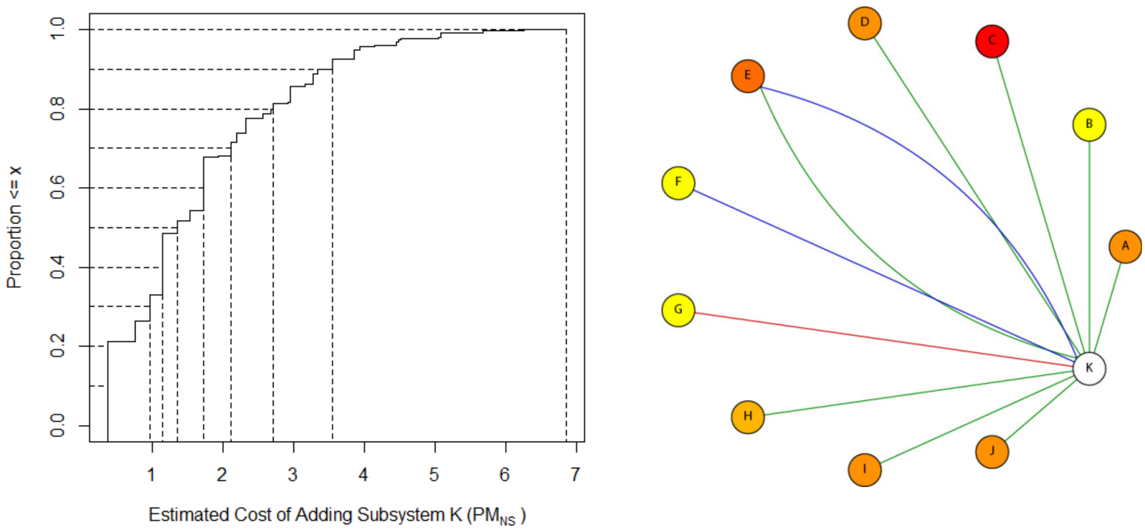


Fig. 4. (a) The empirical cumulative distribution function for the additional cost of adding subsystem K. (b) A graph depicting the frequency and complexity of attachment. The color of the vertices represents the observed, relative frequency that subsystem K attached to each subsystem of the current system, ranging from least frequent (yellow) to most frequent (red); the edge color indicates the most likely complexity of the interface (green ⇒ easy, blue ⇒ nominal, red ⇒ difficult).

Table 4. Deciles for estimated cost of adding subsystem K.

$P(X \leq x)$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
x	0.38	0.38	0.97	1.15	1.36	1.74	2.12	2.71	3.55	6.85

4.3. Limitations

As with any model, this approach has shortcomings. For example, one might question the use of the current degree distribution to model the number of interfaces for subsystem K. After all, there is a noticeable gap in the probability mass function (pmf) of D_{vK} between $d = 4$ and 7, suggesting that subsystem K would never have 5 or 6 interfaces. Moreover, with only 10 subsystems in the current network, one might argue that we have a small sample from which to build the pmf in the first place. On the first point, we could modify the pmf to place a small amount of mass on $d = 5$ and 6, and some risk analysis software provides this functionality (notably Vose's ModelRisk) [21]. This is worth exploring. As for the second point, we argue that we do not have a sample at all, as we are using all the vertices in the network. Accordingly, our 10 subsystems are the population, albeit a small one.

More subtly, the addition of subsystem K could reasonably necessitate the "rewiring" of the existing architecture. For instance, as seen in Figure 3a, connecting subsystem K to A and E could plausibly eliminate the need for the existing, moderately complex interface between A and E. Similarly, if K connects to B, then B may have to connect to C. The addition and removal of edges to an existing network is not new [22], and it holds promise for further refinements to the model. That said, adjusting the current architecture in response to the addition of a generic subsystem should likely be treated as a higher order effect.

Lastly, the presumption that we know nothing about the purpose or function of subsystem K is questionable. More likely than not, the engineers involved in the development of the system would have information regarding potential improvements or refinements, and this knowledge would help refine our estimate. That said, the model presented in this paper is meant to be a very general example of how network science, in conjunction with MSBE and parametric cost modeling, holds promise for future research and implementation.

5. Future Work

The next steps in this research include further refinement of the methodology presented to be able to validate its utility on a real system. This will uncover potential refinements, one of which we anticipate will be the incorporation of reuse. That is, the introduction of subsystems that have already been developed will be less expensive than introducing new subsystems.

Once the methodology is complete, we plan to implement it in a software package so that it can be disseminated to organizations interested in performing these types of sensitivity analyses. As DoDAF and cost models broaden their scope, our methodology can be adapted to accommodate an increasing number of rules. Such capabilities will help mature MBSE and make systems engineering more rigorous in its use of tools and models.

References

1. Friedenthal, Sanford, Regina Griego, and Mark Sampson, INCOSE Model Based Systems Engineering (MBSE) Initiative, 17th INCOSE Symposium, San Diego, CA (2007).

2. Hause, M., D. Richards, and A. Stuart, Testing Solutions through SysML/UML, 19th INCOSE Symposium, Singapore (2009).

3. Saunders, S., Does a Model Based Systems Engineering Approach Provide Real Program Savings?, presented at Informal Symposium on Model-Based Systems Engineering DSTO, Edinburgh, South Australia (2011).

4. Wennergren, D. M., The Department of Defense Architecture Framework (DoDAF) Version 2.0, DoD CIO Memo, Washington, D.C. (May 28, 2009).

5. Bulluss, G. and K. O'Shea, Architecture-based Decision Support for Capability Development, presented at the Australian Society for Defence Engineering (ASDE) (Feb. 8, 2012).

6. NASA, NASA Cost Estimating Handbook, Washington, D.C. (2008).

7. GAO, Cost Estimating and Assessment Guide: Best Practices for Developing and Managing Capital Program Costs, GAO-09-3SP (Mar.

2009).

8. GAO, Lack of Disciplined Cost-Estimating Procedures Hinders Effective Program Management, GAO-04-642 (May 2004).

9. GAO, DoD Needs to Take More Action to Address Unrealistic Initial Cost Estimates of Space Systems, GAO-07-96 (Nov. 2006).

10. Valerdi, R, *The Constructive Systems Engineering Cost Model (COSYSMO): Quantifying the Costs of Systems Engineering Effort in Complex Systems*, Saarbrücken, Germany, VDM Verlag (2008).

11. Wang, G., R. Valerdi, G. Roedler, A. Ankrum, and J. Gaffney, Harmonizing Software Engineering and Systems Engineering Cost Estimation, *International Journal of Computer Integrated Manufacturing* 25(4-5) (2012) 432-443.

12. DoD, DoD Architecture Framework Version 2.0: Volume 2 - Architectural Data and Models Architect's Guide, (May 28, 2009).

13. Barabási, Albert-László, *Linked: How Everything Is Connected to Everything Else and What It Means*, New York, NY: The Penguin Group, 2003.

14. Robinson, Santiago B, A Modeling Process to Understand Complex System Architectures, PhD Dissertation, Georgia Institute of Technology (Aug. 2009), retrieved from SMARTech web site: http://smartech.gatech.edu/handle/1853/29621.

15. Kreimeyer, Matthias and Udo Lindemann, *Complexity Metrics in Engineering Design: Managing the Structure of Design Processes*, New York, NY: Springer-Verlag, 2011.

16. Garret, Robert K., Jr., Steve Anderson, Neil T. Baron, and James D. Moreland, Jr., Managing the Interstitials, a System of Systems Framework Suited for the Ballistic Missile Defense System, *Systems Engineering* 14(1) (2010) 87-109.

17. Csárdi, Gábor, and Tamás Nepusz, The igraph software package for complex network research, *InterJournal* Complex Systems (2006).

18. Barabási, Albert-László and Réka Albert, Emergence of Scaling in Random Networks, *Science* 286 (Oct. 19, 1999) 509-512.

19. Cooper, Colin and Alan Frieze, General Model of Web Graphs, *Random Structures & Algorithms* 22(3) (Mar. 10, 2003) 311-335.

20. Deijfen, Maria, Henri van den Esker, Remco van der Hofstad, and Gerard Hooghiemstra, A preferential attachment model with random initial degrees, *Arkiv för Matematik* 47(1) (2009) 41-72.

21. Vose Software, Discrete fitted distribution, retrieved from Vose Software web site: http://www.vosesoftware.com/ModelRiskHelp/index.htm#Distributions/Discrete_distributions/Discrete_Fitted_distribution.htm.

22. Albert, Réka and Albert-László Barabási, Statistical mechanics of complex networks, *Reviews of Modern Physics* 74 (Jan. 2002) 47-97.